

3. ANALISIS DAN DESAIN SISTEM

Pada bab ini akan dibahas lebih lanjut mengenai analisis dan desain daripada sistem yang akan dikerjakan pada tugas akhir ini. Analisis mencakup permasalahan serta sistem atau perangkat yang akan digunakan untuk melakukan simulasi.

3.1. Analisis Permasalahan dan Kebutuhan

Seperti yang sudah diketahui dari latar belakang masalah yang telah diuraikan pada bab sebelumnya, terlihat jelas bahwa kemacetan di kota Surabaya ini sudah merupakan salah satu isu krusial yang mendesak untuk segera diatasi. Kondisi saat ini adalah di beberapa kawasan atau titik persimpangan di Surabaya menunjukkan bahwa durasi lampu lalu lintas yang ada tidak selaras dengan volume lalu lintas yang sebenarnya. Hal inilah yang sering kali dinilai menjadi titik kritis penyebab adanya penumpukan kendaraan pada persimpangan. Pengaturan lampu lalu lintas yang berlaku saat ini pun dirasa masih belum cukup efektif sehingga berakibat pada arus lalu lintas yang tidak lancar dan adanya peningkatan waktu tunggu yang cukup signifikan bagi para pengendara. Mengingat pentingnya masalah ini, terdapat kebutuhan untuk mengembangkan algoritma dari perhitungan durasi lampu lalu lintasnya. Perhitungan yang dibutuhkan seharusnya bisa menyesuaikan kondisi aktual dari jalan sehingga durasi yang dihasilkan bisa lebih optimal dan sesuai.

Berdasarkan data kondisi jalan yang telah diperoleh, sistem akan memproses dan menyesuaikan pengaturan lampu lalu lintas sesuai dengan kondisi aktual di lapangan, kemudian melakukan pembaruan pengaturan setiap frekuensi waktu yang ditentukan atau secara periodik. Terdapat berbagai faktor kondisi jalan yang harus dipertimbangkan dalam pengembangan sistem pengaturan lampu lalu lintas ini, agar dapat mencapai tingkat efisiensi yang maksimal. Mengingat tidak memungkinkannya melakukan pengujian sistem secara langsung di jalan raya kota Surabaya, maka pengujian pengembangan ini akan dilaksanakan melalui simulasi transportasi. Simulasi nantinya akan dilakukan dengan bantuan *tools Simulation of Urban MObility* atau SUMO. Studi kasus yang akan digunakan dalam simulasi ini adalah jaringan jalan di sekitar Universitas Kristen Petra Surabaya yakni Jl. Siwalankerto.

Akan disiapkan beberapa skenario pengujian yang berbeda untuk memastikan sistem dapat bekerja dengan baik dalam berbagai kondisi. Data yang digunakan dalam simulasi ini akan disesuaikan dengan data pengamatan lapangan. Tetapi, terdapat juga keterbatasan dari simulator yang digunakan yang dimana simulator tidak bisa mensimulasikan beberapa kendaraan dalam 1 lajur sehingga akan dilakukan juga pengujian untuk mendapatkan

perbandingan yang tepat yang menyerupai kondisi aktual tentang perilaku jalan pada persimpangan. Sehingga harapannya simulasi dapat menghasilkan hasil yang cukup akurat karena disesuaikan dengan kondisi sesungguhnya yang ada di Surabaya.

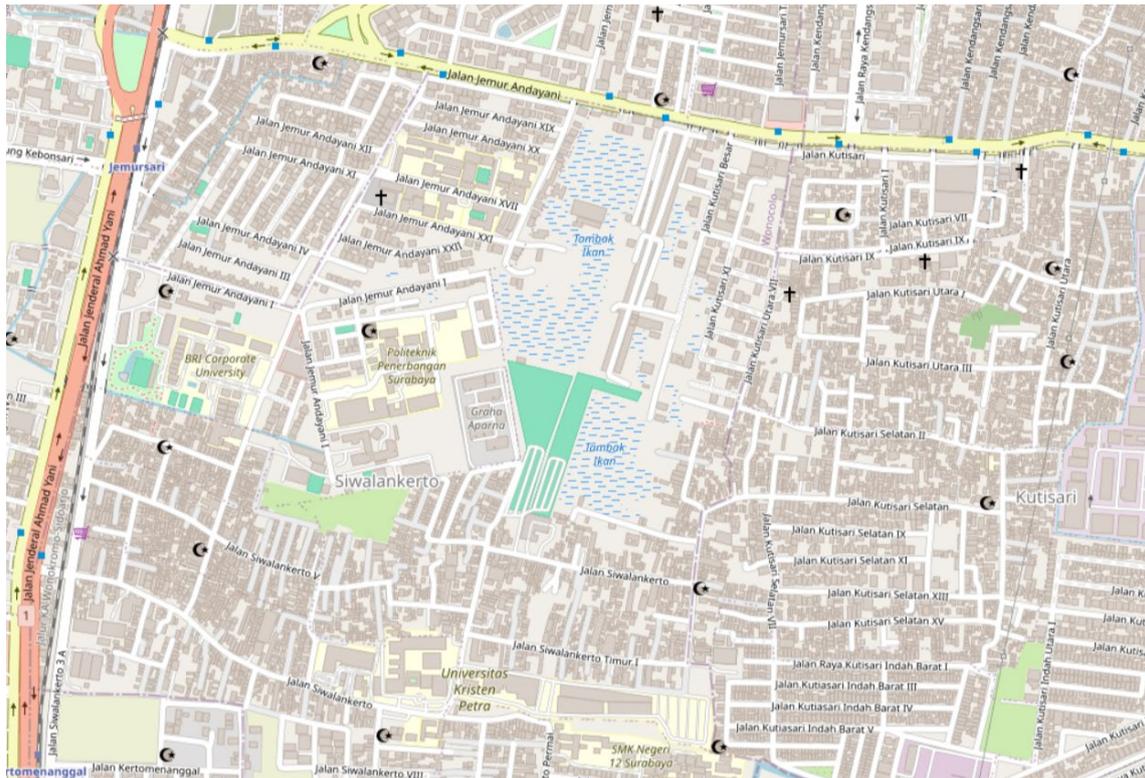
3.2. Analisis Sistem

Untuk dapat membentuk kerangka kerja pengembangan algoritma durasi lampu lalu lintas ini, diperlukan beberapa aspek penting yang akan menjadi fondasi daripada kerangka kerja sistem ini. Aspek-aspek tersebut meliputi pemetaan jalan yang komprehensif, penentuan *input* beserta parameter yang akurat, penggunaan rumus yang tepat, hingga simulasi yang efektif.

Pemetaan Jalan

Proses pemetaan jalan merupakan langkah awal yang krusial pada sistem ini. Pemetaan jalan meliputi identifikasi jaringan jalan yang akan digunakan. Pemetaan inilah yang akan menjadi representasi visual dari jaringan jalan. Dalam kasus ini, akan meliputi jaringan Jl. Siwalankerto Surabaya dan sekitarnya. Nantinya pada proses pemetaan akan digambarkan peta jaringannya. Kemudian pengaturan kondisi jalan, mulai dari jumlah lajur, arah pergerakan, dan interaksi antara jenis kendaraan yang ada pada jalan.

Pemetaan nantinya dapat dilakukan dengan beberapa cara, bisa digambarkan secara manual dengan menggunakan *tools* yang ada pada SUMO simulator, yaitu *netedit* yang berupa file XML kemudian di konversi menggunakan *netconvert* agar dapat dibaca pada SUMO simulator. Selain membuat sendiri jaringan jalan, bisa juga menggunakan data dari OpenStreetMap yang kemudian bisa disesuaikan lagi beberapa pengaturannya sesuai dengan kondisi jalan sesungguhnya. Proses pemetaan ini sendiri penting khususnya untuk memahami struktur serta dinamika lalu lintas.



Gambar 3.1. Gambar Jaringan Jalan yang digunakan

Input dan Parameter

Terdapat beberapa input beserta parameter yang akan digunakan pada sistem ini, yaitu sebagai berikut:

- Jumlah kendaraan: jumlah kendaraan yang dimaksud di sini adalah jumlah kendaraan yang melewati titik tertentu dalam periode tertentu. Untuk data ini dapat diperoleh dari sensor yang terpasang pada jalan atau sumber lainnya.
- Jenis kendaraan: jenis kendaraan dapat mempengaruhi kecepatan serta volume lalu lintas.
- *Response time*: dalam hal ini yang dimaksudkan adalah berapa lama waktu atau *delay* yang dibutuhkan oleh kendaraan dari berhenti hingga jalan.
- Kapasitas jalan: penting untuk mengetahui jumlah maksimum kendaraan yang dapat ditangani suatu jalan pada periode waktu yang sama tanpa menyebabkan kemacetan. Hal ini dapat berdasarkan banyak lajur, lebar jalan, dan lain-lain.
- Waktu: penting untuk mengetahui waktu-waktu kritis yang ada pada lalu lintas persimpangan, misalnya, apakah sekarang merupakan jam sibuk (*rush hour*) atau bukan.
- Kondisi cuaca: kondisi cuaca juga bisa mempengaruhi perilaku pengendara, sehingga penting juga untuk mengetahui kondisi cuaca saat ini.

Rumus atau Persamaan

Terdapat beberapa rumus atau persamaan perhitungan dari *input* dan parameter yang akan digunakan pada sistem ini untuk mendapatkan *output* yang diinginkan.

Kapasitas Jalan Perkotaan

Kapasitas atau daya tampung jalan merupakan kemampuan dari suatu ruas jalan untuk menampung arus atau volume lalu lintas yang ideal dalam satu waktu tertentu. Kapasitas dinyatakan oleh jumlah kendaraan yang melewati suatu jalan tertentu dalam satu jam (*passenger car unit/hour*) atau (satuan mobil penumpang/jam). Menurut buku pedoman dari Direktorat Jenderal Bina Marga (2023), untuk dapat mengetahui atau menghitung kapasitas jalan dapat digunakan persamaan di bawah ini:

$$C = C_0 \times FC_{LJ} \times FC_{PA} \times FC_{HS} \times FC_{UK} \quad (3.1)$$

Di mana :

- **C** adalah kapasitas segmen jalan yang sedang diamati, dengan satuan SMP/jam. Jika kondisi segmen jalan berbeda dari kondisi ideal, maka nilai C harus dikoreksi berdasarkan perbedaan terhadap kondisi idealnya dari lebar lajur atau jalur lalu lintas (FC_{LJ}), pemisahan arah (FC_{PA}), kelas hambatan samping (KHS) pada jalan berbahu atau tidak berbahu (FC_{HS}), dan ukuran kota (FC_{UK}).
- **C_0** adalah kapasitas dasar kondisi jalan yang ideal (smp/jam).
- **FC_{LJ}** adalah faktor koreksi kapasitas akibat perbedaan lebar lajur atau jalur lalu lintas.
- **FC_{PA}** adalah faktor koreksi kapasitas akibat Pemisahan Arah lalu lintas (PA) dan hanya berlaku untuk tipe jalan tak terbagi.
- **FC_{HS}** adalah faktor koreksi kapasitas akibat kondisi KHS pada jalan yang dilengkapi bahu atau dilengkapi kereb dan trotoar dengan ukuran yang tidak ideal.
- **FC_{UK}** faktor koreksi kapasitas akibat ukuran kota yang berbeda dengan ukuran kota ideal.

Dari buku pedoman dari Direktorat Jenderal Bina Marga (2023), terdapat nilai-nilai dari kapasitas dasar dan juga faktor koreksi seperti pada tabel-tabel berikut:

Tabel 3.1.

Kapasitas Dasar

Tipe Jalan	C_0 (SMP/jam)	Catatan
4/2-T, 6/2-T, 8/2-T atau Jalan satu arah	1700	Per lajur (satu arah)
2/2-TT	2800	Per dua arah

Tabel 3.2.

Faktor Koreksi Kapasitas Akibat Perbedaan Lebar Lajur

Tipe Jalan	L_{LE} atau L_{JE} (m)	FC_L
4/2-T, 6/2-T, 8/2-T atau jalan satu-arah	$L_{LE} = 3,00$	0,92
	3,25	0,96
	3,50	1,00
	3,75	1,04
	4,00	1,08
2/2-TT	$L_{JE2\text{ arah}} = 5,00$	0,56
	6,00	0,87
	7,00	1,00
	8,00	1,14
	9,00	1,25
	10,00	1,29
	11,00	1,34

Tabel 3.3.

Faktor Koreksi Kapasitas Akibat PA pada Tipe Jalan Tak Terbagi

PA %-%	50-50	55-45	60-40	65-35	70-30
FC_{PA}	1,00	0,97	0,94	0,91	0,88

Tabel 3.4.

Faktor Koreksi Kapasitas Akibat KHS pada Jalan dengan Bahu

Tipe jalan	KHS	FC_{HS}			
		Lebar bahu efektif L_{BE} , m			
		≤ 5	1	1,5	$\geq 2,0$
4/2-T	Sangat Rendah	0,96	0,98	1,01	1,03

Tipe jalan	KHS	FC _{HS}			
		Lebar bahu efektif L _{BE} , m			
		≤5	1	1,5	≥2,0
	Rendah	0,94	0,97	1	1,02
	Sedang	0,92	0,95	0,98	1
	Tinggi	0,88	0,92	0,95	0,98
	Sangat Tinggi	0,84	0,88	0,92	0,96
2/2-TT atau Jalan satu arah	Sangat Rendah	0,94	0,96	0,99	1,01
	Rendah	0,92	0,94	0,97	1
	Sedang	0,89	0,92	0,95	0,98
	Tinggi	0,82	0,86	0,9	0,95
	Sangat Tinggi	0,73	0,79	0,85	0,91

Tabel 3.5.

Faktor Koreksi Kapasitas Akibat KHS pada Jalan Berkereb

Tipe jalan	KHS	FC _{HS}			
		Jarak kereb ke penghalang terdekat sejauh L _{KP} , m			
		≤5	1	1,5	≥2,0
4/2-T	Sangat Rendah	0,96	0,98	1,01	1,03
	Rendah	0,94	0,97	1	1,02
	Sedang	0,92	0,95	0,98	1
	Tinggi	0,88	0,92	0,95	0,98
	Sangat Tinggi	0,84	0,88	0,92	0,96
2/2-TT atau Jalan satu arah	Sangat Rendah	0,94	0,96	0,99	1,01
	Rendah	0,92	0,94	0,97	1
	Sedang	0,89	0,92	0,95	0,98
	Tinggi	0,82	0,86	0,9	0,95
	Sangat Tinggi	0,73	0,79	0,85	0,91

Tabel 3.6.

Faktor Koreksi Kapasitas Terhadap Ukuran Kota

Ukuran kota (Juta jiwa)	Kelas kota/kategori kota		Faktor koreksi ukuran kota, (FC _{UK})
<0, 1	Sangat Kecil	Kota kecil	0,86
0,1-0,5	Kecil	Kota kecil	0,9
0,5-1,0	Sedang	Kota menengah	0,94
1,0-3,0	Besar	Kota besar	1
>3,0	Sangat Besar	Kota metropolitan	1,04

Tabel 3.7.

Pembobotan Hambatan Samping

No.	Jenis hambatan samping utama	Bobot
1	Pejalan kaki di badan jalan dan yang menyeberang	0,5
2	Kendaraan umum dan kendaraan lainnya yang berhenti	1,0
3	Kendaraan keluar/masuk Sisi atau lahan samping jalan	0,7
4	Arus kendaraan lambat (kendaraan tak bermotor)	0,4

Volume atau Kepadatan Lalu Lintas

Untuk menentukan seberapa padat suatu jalan dapat dilakukan perhitungan volume atau kepadatannya berdasarkan perbandingan rasio. Jika rasio yang didapatkan lebih besar daripada 1, maka volume melebihi kapasitas yang mana akan mengarah ke kemacetan. Sebaliknya, jika rasio kurang dari 1, berarti jalan masih memiliki kapasitas untuk menangani volume lalu lintas yang ada. Rasio dapat dihitung dengan persamaan di bawah ini:

$$ratio = \frac{v}{c} \quad (3.2)$$

Di mana:

- **v** adalah jumlah kendaraan yang melewati titik tertentu dalam periode waktu tertentu.
- **c** adalah kapasitas jalan atau jumlah maksimum kendaraan yang dapat ditangani oleh suatu jalan dalam periode waktu yang sama.

Durasi Lampu Lalu Lintas dalam Satu Siklus

Untuk dapat menghitung durasi lampu lalu lintas pada masing-masing jalan dalam satu siklus pengaturan lampu lalu lintas dapat digunakan persamaan-persamaan di bawah. Pertama akan dilakukan perhitungan bobot total untuk seluruh jalan.

$$B = \sum_{i=1}^n b_i \quad (3.3)$$

Di mana b_i adalah bobot dari masing-masing jalan pada persimpangan yang dapat dihitung dengan menggunakan persamaan sebagai berikut:

$$b_i = \sum_{j=1}^n (k_j \times r_j) \quad (3.4)$$

Di mana:

- k_j adalah jumlah kendaraan jenis j , jenis kendaraan yang akan digunakan pada simulasi yaitu mobil, motor, truk, dan bis.
- r_j adalah *response time* dari kendaraan jenis j . *Response time* kendaraan yang dimaksud pada penelitian ini adalah, waktu yang dibutuhkan dari suatu kendaraan ketika berada pada posisi berhenti atau *stand by* hingga jalan. Untuk *response time* yang akan digunakan pada penelitian ini didasarkan dari observasi lapangan. Didapati bahwa mobil memiliki *response time* antara 2 sampai 3 detik, motor 0,5 sampai 1 detik, bus 2 sampai 4 detik, dan truk 3 sampai 5 detik dalam kondisi jalan normal.

Untuk beberapa skenario seperti pengaruh jam sibuk dan cuaca, *response time* dari masing-masing kendaraan dapat berubah-ubah, sehingga pada simulasi nanti akan ada beberapa macam *response time* dari kendaraan yang akan digunakan seperti pada tabel 3.8. di bawah ini:

Tabel 3.8.

Response Time Kendaraan

Kendaraan	<i>Response Time</i> (detik)
-----------	------------------------------

	Normal	Jam Sibuk / Cuaca Buruk
Mobil	2 - 3	4 - 5
Motor	0.5 - 1	2.5 - 3
Bus	2 - 4	4 - 6
Truk	3 - 5	5 - 7

Setelah melakukan perhitungan daripada bobot dari masing-masing jalan pada persimpangan, maka didapati durasi dari fase lampu hijau dari masing-masing jalan dengan persamaan berikut:

$$g_i = \frac{b_i}{B} \times (T - n \times I) \quad (3.5)$$

Di mana:

- g_i adalah durasi lampu hijau dalam detik.
- b_i adalah bobot dari jalan i .
- B adalah bobot total.
- T adalah durasi total dari satu siklus pengaturan lampu lalu lintas.
- n adalah jumlah jalan pada persimpangan.
- I adalah durasi interval (atau bisa saja lampu kuning) untuk setiap pergantian jalan.

Untuk perhitungan durasi per siklusnya didasarkan pada hasil observasi lapangan pada persimpangan yang ada di Surabaya, yang di mana bervariasi mulai dari 45 detik hingga 180 detik, tergantung pada kondisi jalan. Maka dari itu durasi per siklus akan diasumsikan dengan beberapa kondisi, yaitu:

Tabel 3.9.

Durasi Siklus Lampu Lalu Lintas

No.	Kondisi Jalan	Jam Sibuk	Cuaca	Durasi (detik)
1	Sepi	Tidak	Cerah	45
2	Sepi	Tidak	Buruk	60
3	Sepi	Ya	Cerah	75
4	Sepi	Ya	Buruk	90
5	Ramai	Tidak	Cerah	105

No.	Kondisi Jalan	Jam Sibuk	Cuaca	Durasi (detik)
6	Ramai	Tidak	Buruk	120
7	Ramai	Ya	Cerah	135
8	Ramai	Ya	Buruk	150
9	Padat	Tidak	Cerah	165
10	Padat	Tidak	Buruk	180
11	Padat	Ya	Cerah	180
12	Padat	Ya	Buruk	180

Untuk jalan berpalang pintu kereta api, tentu nantinya terdapat waktu tunggu tambahan. Menurut Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 94 Tahun 2018, palang pintu harus sudah mulai menutup paling sedikit 60 detik sebelum kereta api lewat, dan untuk proses buka dan tutup dari palang pintu sendiri membutuhkan waktu 4 sampai 7 detik. (Kementerian Perhubungan Republik Indonesia, 2018, pp. 13, 19). Dari hasil pengamatan juga didapati bahwa durasi rata-rata kereta yang melewati kawasan Jl. Siwalankerto adalah sekitar 40 detik, sehingga untuk simulasi ini akan diasumsikan waktu kereta lewat adalah 40 detik. Oleh karena itu durasi waktu tunggu tambahan untuk kondisi dengan kereta api adalah sebagai berikut:

$$r = 60 + 7 + 40 \quad (3.6)$$

$$r = 107 \text{ detik}$$

Sehingga untuk skenario dengan kereta api maka durasi lampu lalu lintas dalam satu siklus adalah:

$$T = T' + r \quad (3.7)$$

Di mana:

- **T** adalah durasi total dari satu siklus.
- **T'** adalah durasi total dari satu siklus tanpa adanya kereta api.
- **r** adalah durasi waktu tunggu tambahan untuk kereta api yaitu 107 detik.

Arus Lalu Lintas (*Traffic Flow*)

Rumus (3.8) digunakan untuk menghitung arus lalu lintas. Hasil dari perhitungan rumus ini dapat menentukan apakah arus bisa tergolong lancar atau tidak. Untuk dapat mengetahui apakah arus tersebut lancar atau tidak, penting untuk melakukan perbandingan dengan kapasitas dan kondisi jalan.

$$Q = \frac{N}{t} \quad (3.8)$$

Di mana:

- **Q** adalah *flow* atau arus lalu lintas (vehicles/second)
- **N** adalah jumlah kendaraan yang lewat dalam t detik.

Untuk menghitung arus lalu lintas, akan didasarkan pada hasil dari simulasi. Terdapat perbedaan perilaku kendaraan khususnya motor pada GUI simulator dan kondisi sesungguhnya yang ditemui di jalan. Pada simulator satu lajur hanya dapat berisi satu motor. Oleh karena itu untuk mengatasi hal ini, akan dilakukan pengujian untuk menentukan perbandingan yang sesuai terhadap jumlah motor pada simulator dengan yang ada pada kondisi aktual.

Simulasi Sistem

Untuk mensimulasikan sistem pengembangan algoritma durasi lampu lalu lintas ini akan digunakan SUMO (Simulation of Urban MObility). SUMO merupakan simulator *open-source* yang dapat digunakan untuk mensimulasikan berbagai skenario lalu lintas dengan data dari jaringan jalan yang disediakan.

Setup Environment

Untuk dapat melakukan simulasi, terdapat beberapa pengaturan pada *environment* perangkat lunak yang cukup penting untuk dilakukan agar perangkat lunak atau simulator dapat berjalan dengan baik. Berikut adalah beberapa tahapan untuk melakukan pengaturan atau *setup environment*:

- Instalasi Python: Python juga dapat diunduh dari situs web resmi Python. Disini penulis menggunakan Python versi 3.12.0.
- Instalasi Simulation of Urban MObility (SUMO): SUMO dapat diunduh dari situs web resmi SUMO. Setelah itu bisa dilanjutkan dengan mengikuti petunjuk instalasi yang ada. Disini

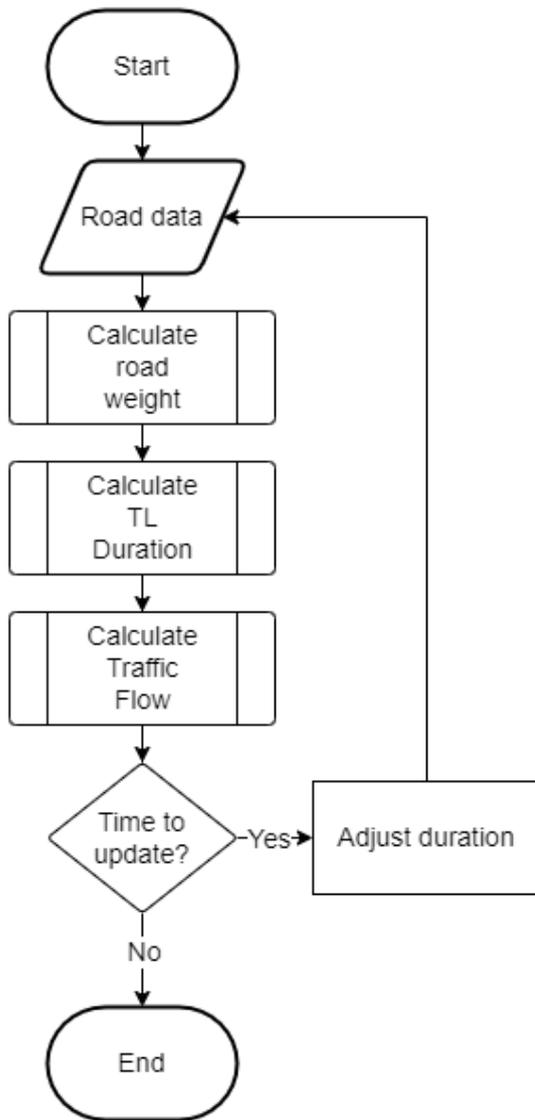
penulis menggunakan simulator SUMO versi 1.19. Setelah instalasi SUMO berhasil perlu dilakukan beberapa penyesuaian pada *environment variable* seperti *path* direktori dari folder instalasi agar nantinya seluruh aplikasi-aplikasi yang terdapat pada paket instalasi SUMO dapat digunakan dengan baik.

Tahapan simulasi

Simulasi dimulai dari tahapan *import* data jaringan jalan yang di mana sudah dibuat pada tahapan pemetaan jalan. Perlu dipastikan bahwa jaringan jalan sudah dikonversi dengan *netconvert* agar bisa terbaca pada simulator (SUMO). Setelah itu bisa dilakukan penyesuaian lagi apabila masih terdapat beberapa aspek yang belum sesuai, seperti *nodes*, *edge*, konektor, informasi rute kendaraan, jadwal transportasi umum (apabila ada), pola-pola perjalanan, dan lain-lain. Karena di sini simulasi digunakan untuk menguji perhitungan sistem, maka akan diasumsikan beberapa skenario pengujian.

Pada SUMO sendiri sudah terdapat beberapa pengaturan dasar atau *default* yang bisa digunakan, seperti pengaturan perhitungan kecepatan rata-rata kendaraan, ataupun volume lalu lintas, nantinya pada saat pengujian pengaturan *default* tersebut juga dapat digunakan sebagai pendukung ataupun pembanding. Pengaturan ini juga bisa disesuaikan sendiri menyesuaikan skenario pengujian dengan mengubah parameter dalam *file* konfigurasi atau menggunakan skrip Python untuk kontrol simulasinya. Beberapa parameter yang dapat dimodifikasi antara lain *range* waktu simulasi, durasi simulasi, jenis kendaraan, perilaku kendaraan, maupun frekuensi generasi daripada kendaraan itu sendiri.

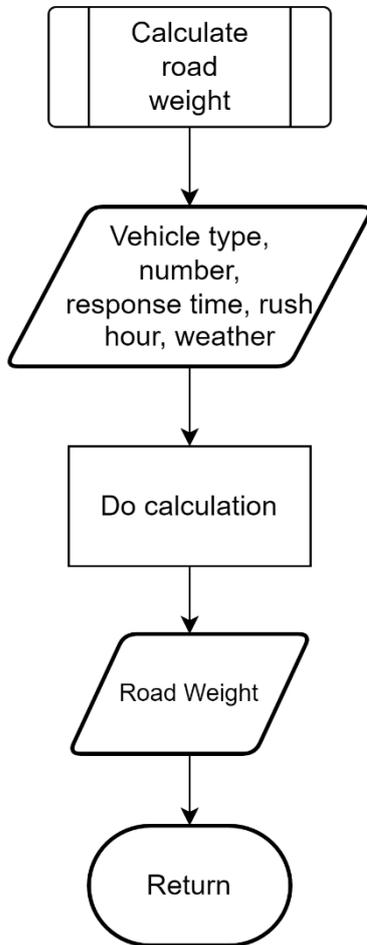
3.3. Desain Sistem



Gambar 3.2. *Flowchart* Desain Sistem

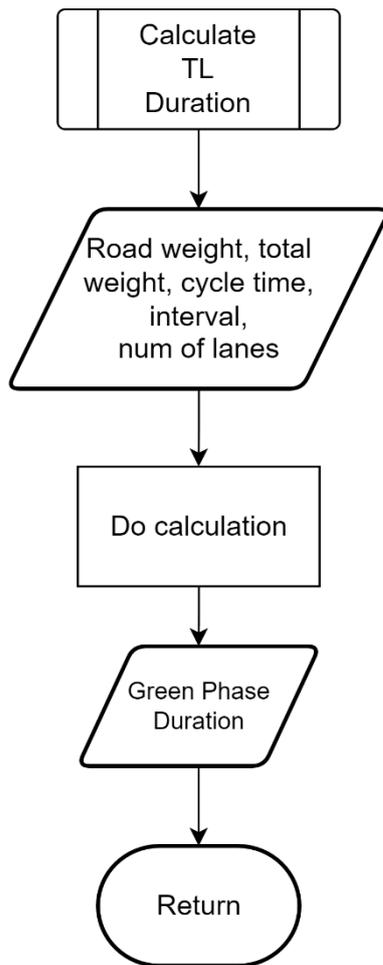
Proses kerja pada Gambar 3.2. *Flowchart* desain sistem, merupakan proses alur kerja sistem dari mendapatkan *input* hingga menghasilkan *output* yang diinginkan. Pertama-tama sistem akan menerima input berupa data kondisi jalan, dimulai dari kapasitas jalan, informasi kendaraan yang ada, seperti jumlah kendaraan yang melintas ataupun jenis kendaraan, kondisi cuaca, dan waktu perjalanan apakah pada jam sibuk atau tidak. Setelah itu akan dilakukan perhitungan-perhitungan untuk persamaan-persamaan lainnya. Setelah itu, sistem akan mengecek secara berkala, apakah ini sudah waktu untuk melakukan pembaruan pengaturan,

jika iya, maka sistem akan membaca ulang kondisi jalan, serta menghitung ulang untuk persamaan-persamaan yang ada, kemudian melakukan pembaruan terhadap durasi lampu.



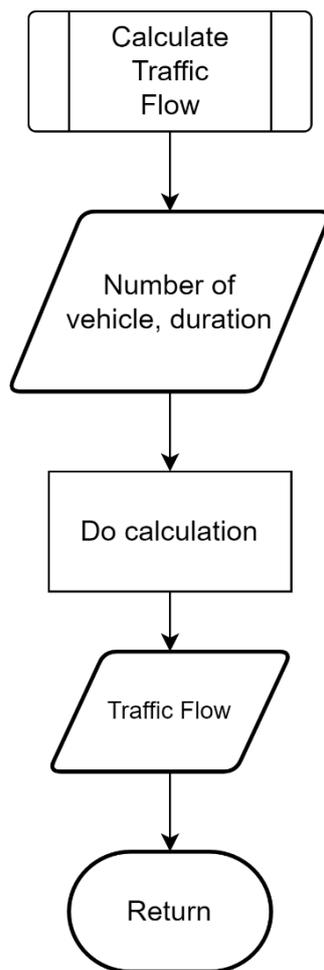
Gambar 3.3. *Flowchart* Perhitungan Bobot Jalan

Pada gambar 3.3. diatas, menunjukkan proses perhitungan untuk menghitung bobot dari jalan. Variabel-variabel *input* seperti jenis kendaraan, jumlah kendaraan, serta *response time* kendaraan sesuai dengan ketentuan apakah jam sibuk atau tidak, dan bagaimana kondisi cuacanya. Kemudian sistem akan melakukan perhitungan sesuai dengan persamaan 3.4. Setelah itu sistem akan mengembalikan bobot jalan sebagai *output*.



Gambar 3.4. Flowchart Perhitungan Durasi Lampu Hijau

Pada gambar 3.4. diatas, menunjukkan proses perhitungan untuk durasi lampu hijau. Sistem menerima variabel bobot jalan, bobot total, durasi siklus lampu lalu lintas, interval lampu lalu lintas, serta jumlah lajur sebagai *input*. Untuk bobot total bisa didapatkan dari hasil perhitungan menggunakan persamaan 3.3. Setelah itu sistem akan melakukan perhitungan durasi lampu menggunakan persamaan 3.5. dan mengembalikan nilai durasi lampu hijau yang didapatkan.



Gambar 3.5. *Flowchart* Perhitungan Arus Lalu Lintas

Pada gambar 3.5. di atas, ditunjukkan proses perhitungan untuk arus lalu lintas. Arus lalu lintas dapat dihitung dengan mengetahui jumlah kendaraan yang lewat pada suatu titik pada durasi tertentu, hal ini juga seperti yang ada pada persamaan 3.8. Kemudian nilai hasil perhitungan arus lalu lintas akan dikembalikan.

Proses ini merupakan siklus yang berkelanjutan, di mana sistem secara periodik akan menyesuaikan pengaturan berdasarkan kondisi lalu lintas yang terus berubah, tetapi untuk pembaruan pengaturan yang dilakukan akan dilakukan sesuai faktor pembaruan yang sesuai. Misalnya, pada penelitian ini dilakukan pembaruan setiap 15 menit. Untuk mengevaluasi sistem ini pun, nantinya akan dilakukan perbandingan kapasitas jalan dengan arus lalu lintasnya. Apabila kapasitas jalan lebih besar daripada arus lalu lintas, maka arus lalu lintas dapat dikatakan lancar.